Wireless Factory ApS
Vestergade 2B, 1 sal
DK-1437 København K
Office: +45 70 20 12 92
www.wirelessfactory.dk

HTTP Gateway

Specification

13. december 2012

Version 2.1.1

Table of Content

1. Welcome

1.1 Introduction

This document describes the simple HTTP interface for sending premium and non-premium SMS, as well as receiving SMS and delivery notifications from the SMS-Gateway of Wireless Factory. The interface represents the simplest of ways to interact with the gateway.

1.2 Contact Information

Wireless Factory ApS
Vestergade 2 B
1456 Copenhagen K
Denmark

Phone: +45 70 20 12 92
Open: 9 am to 4 pm

Website: http://wirelessfactory.dk

1.3 Terms and abbreviations

The table below lists some of the commonly used terms and abbreviations in this document.

Term/Abbreviation	Definition
Channel	A channel refers to the number of an MO and/or MT connection, which can be shortcodes, Large accounts or GSM modem numbers.
Customer	The users of the gateway service of Wireless Factory.
MO	Mobile Originated. A message sent from a mobile is referred to as a MO SMS
MT	Mobile Terminated. When sending a message to a mobile, it is referred to as a MT.
Smsc	Refers to the operator of a given country.
WF	Shorthand for Wireless Factory.
SMS	Short Message Service.
SMSC	Short Message Service Center.
WAP	Wireless Application Protocol. Go to http://www.wapforum.org for details.

Table 1: Terms and abbreviations

2. MT messaging – How to send messages

2.1 Requirements

In order to get your system connected to ours, Wireless Factory we needs to open your specific IP address in our firewall. We should have received this during the initial talks.

For sending through us you have been given a Customer ID and either we have set up a service ID or you have created one yourself. Refer to the gateway user manual for how to set up a service key.

Default character encoding must be observed for successful termination of messages.
The default character encoding is set to CP1252

2.2 Gateway servers

The interface takes a list of input-parameters, which can be sent as either HTTP-POST or HTTP-GET. All parameters must be sent exactly as defined in the table below and must be lowercase. The base

URL for invoking the service is:
http://gateway1.dk.wifact.com/gateway/smspush

To ensure access to the gateway in case of some sort of failure the service can also be reached on following failover URL:

http://gateway2.dk.wifact.com/gateway/smspush

In order to identify and verify the customer requesting an MT SMS to be sent, a static IP is required on the sending server(s). Only messages sent from known IP addresses are processed. The default character encoding is set to CP1252, but can be altered by setting the HTTP encoding parameter[1].

[1] Setting the HTTP header: Content-Type: text/html; charset=ISO-8859-4

2.3 Parameters

All MT messages are required to include a content code and a description of its contents. The gateway administrator can specify a standard content code and description for a service, but sometimes each MT message requires an individual code and description. The following parameters can be used to override the default code. However, a service must be pre-approved for this.

When used, both parameters are required. If the service is not pre-approved, the parameters will be ignored and the message will be sent with the standard content code and description.

Callbackurl and messageid is mandatory for all premium services.

Parameters	Opt.	Format	Description
category		Numeric string	The content code describing the contents of the message. Refer to table 2.1 for a list of valid codes. The customer must be pre-approved for specifying this code, otherwise it will be discarded.
category_description		Alphanumeric string Max length: 50	A user-friendly description of the contents of the message. If the **category** parameter is specified, this parameter must be specified as well. Please include description of supplier, contents AND support phone, address. E.g. *Company A/S, Competition, +45 1234 5678* The contents of this field must not exceed 50 characters.
callbackurl		Alphanumeric string	Delivery notifications are sent to this url. A callbackurl is required for receiving delivery notifications.
channel		Numeric string	The channel used for sending the message. Channels are available upon request, and are typically used for premium messages, and when channel specific capabilities are required e.g. special national characters.
messageid	*	Alphanumeric string	The messageid is an external reference to the messages used when sending delivery notifications to the customer. It is the responsibility of the customer to maintain unique messageid's. If a messageid is not defined, no callback with delivery status will be sent. If sending to a list of MSISDN's this field should be empty or contain a list of the same length as 'mobile'. The list must be separated by a comma(,). Each messageid is used for delivery notification for the message sent to the mobile at the same place in the 'mobile' list.
mobile		Numeric string	The MSISDN of the receiver, country code must always be included. It is possible to send a list of MSISDN's, each entry in the list must be separated by a comma (,) and all MSISDN's must include country code. If sending to a list of MSISDN's other fields will be equal, which means that smsc, text,

			channel and so on will have to be the same for all customers. When making request using http GET method, one should be aware of the size limit for requests.
smsc	*	Alphanumeric string	The smsc of the user e.g. the operator to which the user subscribes. Specifying the smsc of a user is necessary when sending premium MT messages. Please refer to Operator names.
price		Numeric string	The price of the message. Customers are explicitly given access to price categories for individual channels by Wireless Factory. If no price is defined the message is sent zero charged. The price is in the lowest local currency of the channel e.g. Euro Cent. A price of 100 for a Danish premium channel would imply a premium of 1.00 DKK.
senderalias	*	Alphanumeric string Max length: 12	The sender address of the message. Note that not all characters are usable, and that it is not possible to reply to alphanumerical addresses. It is not legal to change the sender of a charged message
servicekey		Numeric string	Supplied by Wireless Factory, can be used to distinguish services in our statistics interface.
loggroup		Alphanumeric string	Used to group MT messages, with same servicekey, in the statistics interface. I.e. set this parameter to the keyword of the MO SMS. Then the MO SMS will be grouped with the MT SMS.
text		Alphanumeric string Max length: 160 before URL-encoding.	The url-encoded text to send. Read section X on SMS length to see how message length is calculated.
timestamp	*	Date string of the format 'YYYYMMDDHH mm'	The message is sent when the given timestamp is reached. Our servers are synchronized with GMT+1. We recommend synchronizing the sending server(s) before using this parameter (e.g. with ntp).
Type	*	Alphanumeric string	If the type is set, the message is sent as a Nokia udh message. The following types are supported: • "ringtone" - Nokia ringtone • "picture" - Nokia picture message • "logo" - Nokia logo ***Deprecated***
Wapurl	*	Alphanumeric string	If defined, the wapurl is sent as a service message to the handset.
customerid		Alphanumeric string	If the customer does not have a static IP address, the customerid must be used as identification of the customer. If the customer can give an ip range this will still be used as security check. We strongly recommend not relying solely on the customerid as security.

donation		0 or 1	If donation is set to 1. The message is tried delivered as a donation message.
paymenttype	*	Alphanumeric string	Parameter indicating payment type. Possible values: • mpayment • donation

Table 2: Parameters MT messaging

Code examples and examples of request parameters are provided in the Examples section.

2.1 Valid content codes

Code	Description
01	Traditional mobile content
02	Video
03	Erotic and other adult content
04	Music
05	Audio
06	Games
07	Chat & Dating
08	Competitions, voting and polls
09	mCommerce
10	News & information
11	Charity and donations
12	B2B
13	Other

2.2 Status Messages

Once the request has been submitted to the gateway, the request will be processed and an HTTP response code will be returned. The HTTP response codes and their associated meaning are listed in the table below.

Http response code	Http response text	Meaning
200	N/A	Message received successfully by gateway.
500	N/A	An internal gateway error occurred. Please contact WF for further info.
403	Unauthorized IP	The request came from an unauthorized IP.
400	Smsc missing or malformed	If the value of smsc parameter is unknown, or the smsc is missing in conjunction with sending a premium message.
400	Error decoding text	Both *text* and *wapurl* is missing, or we are unable to decode the *text* field.
401	Supplied category description is not valid	The parameter **category_description** was either not specified or exceeded its maximum length. This error will only occur if the parameter **category** is specified.
401	Given category <X> is not allowed.	The content code provided (<X>) is not a valid content code (refer to table XXX for a list of valid codes).

400	Malformed mobile	Mobile must be digits only
400	Missing mobile	The *mobile* parameter is mandatory, but not present in the request.
400	Missing channel	When sending premium messages, the customer must supply a channel.
400	Missing messageid	When callback is requested by defining a callbackurl, the customer must supply a unique id to use when making the callback
400	Missing servicekey	The *servicekey* parameter is mandatory, but missing in the request.
400	Missing parameter for callback	Both messageid and callbackurl must be present, but only one is in the request.
401	Price/Channel not allowed for: xxxxxxxx.	The customer has not been approved for this price on this channel.
401	Illegal servicekey	The customer tries to send a message on illegal servicekey.
400	Wrong timestamp format. Should be: yyyyMMddHHmmss	Error in timestamp format.
400	Different number of mobiles and messageids	Delivery notification request when sending to more than one MSISDN requires the same number of messageid's as mobile numbers.

Table 3: HTTP response codes - MT messaging

If an error occurs while pushing to a list of receivers, the http status message will contain two lines, the last will be:

Sent: <count>

Where <count> is the number of messages which were successfully sent, messages are always sent in the order they are placed in the list.

2.3 Delivery notification

The gateway provides delivery notification for messages, which are sent with a messageid and a callbackurl. The delivery notification sent to the customers indicates whether the message has been deliver delivered to the handset or is undeliverable. The parameters which are sent to the callbackurl are described in the table below, as well as the meaning of the possible status codes.

Parameter	Description
messageid	The id matching the id used when sending the message.
status	A code indicating the result.

Table 4: Delivery notification parameters

Status	Meaning
0	Message delivered to handset
1	Message cannot be routed. The possible reasons for this message are: • The specified smsc might be unknown on the specified channel • The channel does not exist. • The customer does not have rights to use the channel.
2	Unknown subscriber – The operator does not recognize the number
3	Account insufficient – The customer has insufficient credit
4	Unknown SMSC – unrecognized operator
5	Timed out
6	Unspecified error
7	Blacklisted – The customer is barred by own request or by operator. Customer needs to contact his operator.
8	The message is delivered to the SMSC, but no further delivery is forthcoming.

Table 5: Delivery notification status code

The various networks support delivery notifications in varying degree, ranging from full support to none. E.g. some operators/channels can only deliver a subset of the status codes above, whilst other do not support delivery notification at all.

3. MO messaging – Receiving messages

3.1 Parameters

The gateway delivers MO messages to a customer defined URL with the following parameters:

Parameter	Opt.	Format	Description
channel		Numeric string	The channel that the message was received on.
mobile		Numeric string	The MSISDN of the user, which is prefixed with the country code.
network	*	Alphanumeric string	The smsc of the user. The smsc is not accessible when the message is received by a modem.
price	*	Numeric string	The MO price of the message. This parameter is only defined for MO-Premium channels. The price is in the lowest local currency of the channel e.g. Euro Cent. A price of 100 for a Danish premium channel would imply a premium of 1.00 DKr.
serviceid		Numeric string	The id of the service to which the message is registered in the gateway. The servicekey is contingent of the channel and/or the message content.
text		Alphanumeric string Max length: 160 before URL-encoding.	The text received from the handset, delivered as cp1252 encoded text
receivetime		yyyyMMddhhmmss	Time WF received the message from the operator. Example: 20111020152108

Table 6: Parameters - MO messaging

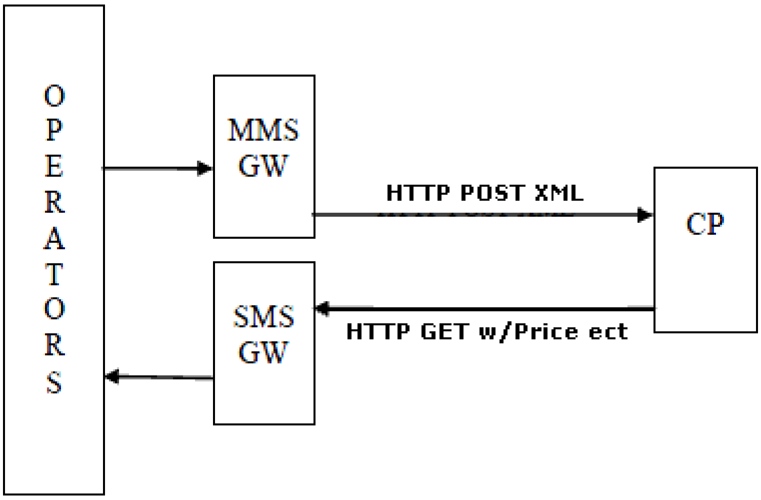
4. MMS MO Messaging

4.1 Introduction

The following describes the simple MMS interface for receiving MMS's through Wireless Factory MMS Gateway. The interface represents the simplest of ways to interact with the gateway.

NOTE: At this point the gateway only serves MO MMS.

4.2 MMS gateway usage scenario



1. When the MMS Gateway receives a MMS from one of the operators, it constructs a XML page. This page gets posted to a Content Provider using HTTP POST.

2. When the Content Provider has received the MMS, the Content Provider should send a MT SMS to WF's SMS Gateway with correct billing info so that the received MMS will be charged.

3. The SMS Gateway will deliver the SMS to the operator with correct billing info, and the user will be charged.

4.3 XML Tags

The Gateway delivers MO MMS as XML in a HTTP POST to a customer defined URL.

The Following tags are to be found inside the XML document.

Tag	Opt.	Description
mo		Root tag describing which type of MMS it is (MO or MT).
msgid		A unique id given by the gateway identifying the message.
mobile		The MSISDN of the user, which is prefixed with the country code.
network		Which network the MO message came from. (tdc, sonofon, etc.)
channel		Which shortcode the message was sent to.
keyword		Which keyword the message was sent to.
message		Which type of message this is. Can be either SMIL or MMS
timestamp		Time when the message was received in gateway.
subject		Subject of the message.
content		Tag holding content part(s) of the message.
presentation	*	Tag holding URL of SMIL presentation file. Only available if message is set to SMIL
text	*	Tag holding URL of text file
image	*	Tag holding URL of image file.
sound	*	Tag holding URL of sound file.
video	*	Tag holding URL of video file.

Table 2: Tags - MO messaging

4.4 Example MO type MMS

```
<?xml version="1.0" encoding="iso-8859-1"?>
<mo>
  <msgid>654321</msgid>
              <mobile>4512345678</mobile>
              <network>tdc</network>
              <channel>1220</channel>
              <keyword>wfmms</keyword>
              <message type="mms">
                <timestamp>2005-02-01 12:00:00</timestamp>
              <subject>Check this picture</subject>
              <content>
                <text>
                  http://mms.wirelessfactory.dk/mms/mo/654321/picture.txt
                </text>
                <image>
                  http://mms.wirelessfactory.dk/mms/mo/654321/pix.jpg
                </image>
              </content>
  </message>
</mo>
```

4.5 Example MO type SMIL

```
<?xml version="1.0" encoding="iso-8859-1"?>
<mo>
  <msgid>654321</msgid>
              <mobile>4512345678</mobile>
              <network>tdc</network>
              <channel>1220</channel>
              <keyword>wfmms</keyword>
              <message type="smil">
                <subject>Check this Presentation</subject>
                <content>
                  <presentation>
                    http://wirelessfactory.dk/mms/mo/654321/mms.smil
                  </presentation>
                  <text>
                    http://wirelessfactory.dk/mms/mo/654321/picture.txt
                  </text>
                  <image>
                    http://wirelessfactory.dk/mms/mo/654321/pix.jpg
                  </image>
                </content>
              </message>
</mo>
```

5. Example – Messages

5.1 Incoming MO message

Incoming message will look like:

mobile=45xxxxxxxx&channel=1220&network=tdc&text=keyword+more+text&serviceid=yyyyyyy

5.2 Example MT Premium

Sending a normal text message to MSISDN 45xxxxxxxx for service yyyyyyyy:

http://gateway1.wifact.com/gateway/smspush?customerid=000000000000000&mobile=45xxxxxxxx&text=hello+world&serviceid=yyyyyyyy&channel=1220

The same message charged 1 DKK sent to a TDC customer:

http://gateway1.wifact.com/gateway/smspush?customerid=000000000000000&mobile=45xxxxxxxx&text=hello+world&serviceid=yyyyyyyy&channel=1220&price=100&smsc=tdc

A donation would be sent as:

http://gateway1.wifact.com/gateway/smspush?customerid=000000000000000&mobile=45xxxxxxxx&text=hello+world&serviceid=yyyyyyyy&channel=1220&price=2500&smsc=tdc&donation=1

5.3 Example MT with notification

Notification is requested by adding the parameters 'callbackurl' and 'messageid':

http://gateway1.wifact.com/gateway/smspush?customerid=000000000000000&mobile=45xxxxxxxx&text=hello+world&serviceid=yyyyyyyy&channel=1220&price=100&smsc=tdc&callbackurl=http://yoursite.com/service&messageid=45xxxxxxxx:uniqueID

We will deliver notification like:

messageid=45xxxxxxxx:uniqueID&status=0

5.4 Example MT Premium Reserve Capture

5.5 MT mPayment

5.6 MT mPayment Reserve capture

5.7 MT Donation

5.8 MT Bulk

6. Example Code

Following code snippets are only examples, and should not be used to without modifying to you specific needs.

6.1 Java

A method making a URL request to our server could look like below; the String argument is a URL with all argument:

```
public void send(String message){
   try{
      URL url = new URL(message);
      conn = (HttpURLConnection)url.openConnection();
      resp = conn.getResponseCode();

      if(resp != 200){
         System.out.println("Unable to deliver message due to: "+resp+"
         "+conn.getMessage());
      }else{
         System.out.println("Message delivered");
      }
   }catch(Throwable t){
      System.out.println("Unable to make url request: "+t.getStackTrace());
   }
}
```

6.2 php

In php following is one way to make a request.

```
$url1 = "gateway1.wifact.com";
$url2 = "gateway2.wifact.com";
$str =
"customerid=000000000000000&mobile=45xxxxxx&smsc=tdc&serviceid=xxxxxxxx&text=god+jul&c
hannel=1299";
$header = "POST /gateway/smspush HTTP/1.1\r\n";
$header .= "Host: gateway1.wifact.com\r\n";
$header .= "Content-type: application/x-www-form-urlencoded\r\n";
$header .= "Content-length: ".strlen($str)."\r\n\r\n";
$header .= $str;
$fp1 = fsockopen($url1,80,$errno,$errstr,10);

if ($fp1) {
   fputs($fp1,$header);
}
else {
   $fp2 = fsockopen($url1,80,$errno,$errstr,10);
   if ($fp2) {
      fputs($fp2,$header);
  }
}
```

7. Operator names

A list of available operator names in Denmark.

We are aware that Sonofon has changed name to Telenor but the Sonofon name is still used to identify the network technically.

If needed Wireless Factory can provide an interface (OCH) where customers can lookup operator for a given MSISDN. Please make inquiry to Wireless Factory to obtain details.

Operator
tdc
sonofon
telia
hi3g

Table 7: Operators

8. Approved tariffs

8.1 mBilling - Digital premium services

The following tariffs[2] are approved when billing digital premium services.

DKK	DKK	DKK
0	19	99
0,5	20	100
1	21	109
1,5	22	110
2	23	119
2,5	24	120
3	25	129
3,5	26	130
4	27	139
4,5	28	140
5	29	149
5,5	30	150
6	35	159
6,5	39	160
7	40	169
7,5	45	170
8	49	179
8,5	50	180
9	55	189
9,5	59	190
10	60	199
11	65	200
12	69	
13	70	
14	75	
15	79	
16	80	
17	89	
18	90	

[2] All tariffs are inc. VAT.

8.2 mPayment – Physical goods

The following tariffs[3] are approved when billing physical premium services.

DKK	DKK	DKK
0	19	99
0,5	20	100
1	21	109
1,5	22	110
2	23	119
2,5	24	120
3	25	129
3,5	26	130
4	27	139
4,5	28	140
5	29	149
5,5	30	150
6	35	159
6,5	39	160
7	40	169
7,5	45	170
8	49	179
8,5	50	180
9	55	189
9,5	59	190
10	60	199
11	65	200
12	69	209
13	70	210
14	75	219
15	79	220
16	80	225
17	89	
18	90	

[3] All tariffs are inc. VAT.

8.3 mDoantion – Donation services

The following tariffs[4] are approved when billing donation services.

DKK	*DKK*	*DKK*
0	40	110
5	50	120
10	60	130
15	70	140
20	80	150
25	90	
30	100	

[4] All tariffs are VAT exempted.

9. Keywords

A web application that enables the user to
- Check availability of a keyword
- Order a keyword
- Stop a keyword

The requests are only allowed when a correct customer ID is provided.
Please make sure you have a contract with cost pr. keyword before using this part of the API as the cost pr. keyword pr. month without a contract is 100 DKK. Prices can be found in Addendum 1 or 2.

9.1 Keyword availability check

URL: http://log.wirelessfactory.dk/keywordhandler/is_available

Parameters	
customerid	The customer ID provided by Wireless Factory.
keyword	The keyword which to be checked.
shortcode	The shortcode for the keyword.
country	The country code for the short code.

The service returns HTTP status 200 OK if the requests is successful (i.e., if all parameters are valid). There are two possible return values in a successful request:

- 1 = Keyword is available
- 0 = Keyword is unavailable

Error codes	
400 Bad Request	One or more parameters are missing.
403 – 001 Invalid	The provided customer ID is invalid.
403 – 003 Illegal	The provided shortcode is not allowed to be used
403 – 004 No More	The customer has reached its keyword limit on the
500 Internal Server	An internal error has occurred. Please try again

Example:

http://log.wirelessfactory.dk/keywordhandler/is_available?customerid=1&keyword=test&shortcode=1220&country=45

Customer with ID 1 checks if the keyword test on the Danish shortcode 1220 is available.

9.2 Order keyword

URL: http://log.wirelessfactory.dk/keywordhandler/order

Parameters	
customerid	The customer ID provided by Wireless Factory.
servicekey	The service key on which the keyword should be
keyword	The keyword which to be checked.
shortcode	The shortcode for the keyword.
country	The country code for the short code.
url_alias	The name of the URL messages with this keyword should be dispatched to. This name can be found in the gateway administration.
email	The email that should receive updates for this

The service returns HTTP status 200 OK if the requests is successful (i.e., if all parameters are valid). There are two possible return values in a successful request:

- 1 = Keyword has been ordered
- 0 = Keyword has not been ordered (e.g. because it is not available)

Error codes	
400 Bad Request	One or more parameters are missing.
403 – 001 Invalid Customer ID	The provided customer ID is invalid.
403 – 002 Invalid Service Key	The customer does not own the specified service key (if it exists)
403 – 003 Illegal Shortcode	The provided shortcode is not allowed to be used by this customer.
403 – 004 No More Keywords Allowed	The customer has reached its keyword limit on the given shortcode.
403 – 005 Invalid URL Alias	The name of the dispatcher URL is invalid.
500 Internal Server Error	An internal error has occurred. Please try again later.

Example:

http://log.wirelessfactory.dk/keywordhandler/order?customerid=1&&servicekey=2&keyword=test&shortcode=1220&country=45&url_alias=MyUrl&email=test@test.test

Customer with ID **1** on servicekey **2** orders the keyword **test** on the **Danish** shortcode **1220**. Messages will be dispatched to the url called **MyUrl**.

9.3 Delete keyword

URL: http://log.wirelessfactory.dk/keywordhandler/stop

Parameters	
customerid	The customer ID provided by Wireless Factory.
servicekey	The service key on which the keyword should be stopped.
keyword	The keyword that is to be stopped.
shortcode	The shortcode on which the keyword exists.
country	The country code for the short code.

The service returns HTTP status 200 OK if the requests is successful (i.e., if all parameters are valid). There are two possible return values in a successful request:

- 1 = Keyword has been stopped
- 0 = Keyword has not been stopped (e.g. because it was not running)

Error codes	
400 – Bad Request	One or more parameters are missing.
403 – 001 Invalid Customer ID	The provided customer ID is invalid.
403 – 002 Invalid Service Key	The customer does not own the specified service key (if it exists)
403 – 006 Invalid Keyword Specified for Customer ID and Service Key	The specified customer does not own the keyword to be stopped (if it exists).
500 – Internal Server Error	An internal error has occurred. Please try again later.

Example:

http://log.wirelessfactory.dk/keywordhandler/stop?customerid=1&servicekey=2&keyword=test&shortcode=1220&country=45

Customer with ID **1** on servicekey **2** stops the keyword **test** on the **Danish** shortcode **1220**.

10. Accepted characters

Dec	Hex	Character name	Character	CP1252	Comments	Sup?
0	0x00	COMMERCIAL AT	@	0x40	d	YES
1	0x01	POUND SIGN	£	0xA3		YES
2	0x02	DOLLAR SIGN	$	0x24	d	YES
3	0x03	YEN SIGN	¥	0xA5		YES
4	0x04	LATIN SMALL LETTER E WITH GRAVE	è	0xE8		YES
5	0x05	LATIN SMALL LETTER E WITH ACUTE	é	0xE9		YES
6	0x06	LATIN SMALL LETTER U WITH GRAVE	ù	0xF9		YES
7	0x07	LATIN SMALL LETTER I WITH GRAVE	ì	0xEC		YES
8	0x08	LATIN SMALL LETTER O WITH GRAVE	ò	0xF2		YES
9	0x09	LATIN CAPITAL LETTER C WITH CEDILLA	Ç	0xC7		YES
10	0x0A	LINE FEED		0x0A	a,d	YES
11	0x0B	LATIN CAPITAL LETTER O WITH STROKE	Ø	0xD8		YES
12	0x0C	LATIN SMALL LETTER O WITH STROKE	ø	0xF8		YES
13	0x0D	CARRIAGE RETURN		0x0D	a,d	YES
14	0x0E	LATIN CAPITAL LETTER A WITH RING ABOVE	Å	0xC5		YES
15	0x0F	LATIN SMALL LETTER A WITH RING ABOVE	å	0xE5		YES
16	0x10	GREEK CAPITAL LETTER DELTA	Δ	N/A		**NO**
17	0x11	LOW LINE	_	0x5F		YES
18	0x12	GREEK CAPITAL LETTER PHI	Φ	N/A		**NO**
19	0x13	GREEK CAPITAL LETTER GAMMA	Γ	N/A		**NO**
20	0x14	GREEK CAPITAL LETTER LAMBDA	Λ	N/A		**NO**
21	0x15	GREEK CAPITAL LETTER OMEGA	Ω	N/A		**NO**
22	0x16	GREEK CAPITAL LETTER PI	Π	N/A		**NO**
23	0x17	GREEK CAPITAL LETTER PSI	Ψ	N/A		**NO**
24	0x18	GREEK CAPITAL LETTER SIGMA	Σ	N/A		**NO**
25	0x19	GREEK CAPITAL LETTER THETA	Θ	N/A		**NO**
26	0x1A	GREEK CAPITAL LETTER XI	Ξ	N/A		**NO**
27	0x1B	ESCAPE TO EXTENSION TABLE			b	
27 10	0x1B0A	**FORM FEED**		0x0C	a,c,d	YES
27 20	0x1B14	**CIRCUMFLEX ACCENT**	^	0x5E	c	YES
27 40	0x1B28	**LEFT CURLY BRACKET**	{	0x7B	c	YES
27 41	0x1B29	**RIGHT CURLY BRACKET**	}	0x7D	c	YES
27 47	0x1B2F	**REVERSE SOLIDUS (BACKSLASH)**	\	0x5C	c	YES
27 60	0x1B3C	**LEFT SQUARE BRACKET**	[	0x5B	c,d	YES
27 61	0x1B3D	**TILDE**	~	0x7E	c	YES
27 62	0x1B3E	**RIGHT SQUARE BRACKET**	]	0x5D	c,d	YES
27 64	0x1B40	**VERTICAL LINE**	|	0x7C	c	YES
27 101	0x1B65	**EURO SIGN**	€	0x80	c	**NO**
28	0x1C	LATIN CAPITAL LETTER AE	Æ	0xC6		YES
29	0x1D	LATIN SMALL LETTER AE	æ	0xE6		YES
30	0x1E	LATIN SMALL LETTER SHARP S (German)	ß	0xDF		YES
31	0x1F	LATIN CAPITAL LETTER E WITH ACUTE	É	0xC9		YES
32	0x20	SPACE		0x20		YES

Dec	Hex	Character name	Character	CP1252	Comments	Sup?
33	0x21	EXCLAMATION MARK	!	0x21	d	YES
34	0x22	QUOTATION MARK	"	0x3F		YES
35	0x23	NUMBER SIGN	#	0x23	d	YES
36	0x24	CURRENCY SIGN	¤	0xA4		**NO**
37	0x25	PERCENT SIGN	%	0x25	d	YES
38	0x26	AMPERSAND	&	0x26	d	YES
39	0x27	APOSTROPHE	'	0x27	d	YES
40	0x28	LEFT PARENTHESIS	(	0x28	d	YES
41	0x29	RIGHT PARENTHESIS	)	0x29	d	YES
42	0x2A	ASTERISK	*	0x2A	d	YES
43	0x2B	PLUS SIGN	+	0x2B	d	YES
44	0x2C	COMMA	,	0x2C	d	YES
45	0x2D	HYPHEN-MINUS	-	0x2D		YES
46	0x2E	FULL STOP	.	0x2E		YES
47	0x2F	SOLIDUS (SLASH)	/	0x2F	d	YES
48	0x30	DIGIT ZERO	0	0x30		YES
49	0x31	DIGIT ONE	1	0x31		YES
50	0x32	DIGIT TWO	2	0x32		YES
51	0x33	DIGIT THREE	3	0x33		YES
52	0x34	DIGIT FOUR	4	0x34		YES
53	0x35	DIGIT FIVE	5	0x35		YES
54	0x36	DIGIT SIX	6	0x36		YES
55	0x37	DIGIT SEVEN	7	0x37		YES
56	0x38	DIGIT EIGHT	8	0x38		YES
57	0x39	DIGIT NINE	9	0x39		YES
58	0x3A	COLON	:	0x3A	d	YES
59	0x3B	SEMICOLON	;	0x3B	d	YES
60	0x3C	LESS-THAN SIGN	<	0x3C		YES
61	0x3D	EQUALS SIGN	=	0x3D	d	YES
62	0x3E	GREATER-THAN SIGN	>	0x3E		YES
63	0x3F	QUESTION MARK	?	0x3F	d	YES
64	0x40	INVERTED EXCLAMATION MARK	¡	0xA1		YES
65	0x41	LATIN CAPITAL LETTER A	A	0x41		YES
66	0x42	LATIN CAPITAL LETTER B	B	0x42		YES
67	0x43	LATIN CAPITAL LETTER C	C	0x43		YES
68	0x44	LATIN CAPITAL LETTER D	D	0x44		YES
69	0x45	LATIN CAPITAL LETTER E	E	0x45		YES
70	0x46	LATIN CAPITAL LETTER F	F	0x46		YES
71	0x47	LATIN CAPITAL LETTER G	G	0x47		YES
72	0x48	LATIN CAPITAL LETTER H	H	0x48		YES
73	0x49	LATIN CAPITAL LETTER I	I	0x49		YES
74	0x4A	LATIN CAPITAL LETTER J	J	0x4A		YES
75	0x4B	LATIN CAPITAL LETTER K	K	0x4B		YES

Dec	Hex	Character name	Character	CP1252	Comments	Sup?
76	0x4C	LATIN CAPITAL LETTER L	L	0x4C		YES
77	0x4D	LATIN CAPITAL LETTER M	M	0x4D		YES
78	0x4E	LATIN CAPITAL LETTER N	N	0x4E		YES
79	0x4F	LATIN CAPITAL LETTER O	O	0x4F		YES
80	0x50	LATIN CAPITAL LETTER P	P	0x50		YES
81	0x51	LATIN CAPITAL LETTER Q	Q	0x51		YES
82	0x52	LATIN CAPITAL LETTER R	R	0x52		YES
83	0x53	LATIN CAPITAL LETTER S	S	0x53		YES
84	0x54	LATIN CAPITAL LETTER T	T	0x54		YES
85	0x55	LATIN CAPITAL LETTER U	U	0x55		YES
86	0x56	LATIN CAPITAL LETTER V	V	0x56		YES
87	0x57	LATIN CAPITAL LETTER W	W	0x57		YES
88	0x58	LATIN CAPITAL LETTER X	X	0x58		YES
89	0x59	LATIN CAPITAL LETTER Y	Y	0x59		YES
90	0x5A	LATIN CAPITAL LETTER Z	Z	0x5A		YES
91	0x5B	LATIN CAPITAL LETTER A WITH DIAERESIS	Ä	0xC4		YES
92	0x5C	LATIN CAPITAL LETTER O WITH DIAERESIS	Ö	0xD6		YES
93	0x5D	LATIN CAPITAL LETTER N WITH TILDE	Ñ	0xD1		YES
94	0x5E	LATIN CAPITAL LETTER U WITH DIAERESIS	Ü	0xDC		YES
95	0x5F	SECTION SIGN	§	0xA7		YES
96	0x60	INVERTED QUESTION MARK	¿	0xBF		YES
97	0x61	LATIN SMALL LETTER A	a	0x61		YES
98	0x62	LATIN SMALL LETTER B	b	0x62		YES
99	0x63	LATIN SMALL LETTER C	c	0x63		YES
100	0x64	LATIN SMALL LETTER D	d	0x64		YES
101	0x65	LATIN SMALL LETTER E	e	0x65		YES
102	0x66	LATIN SMALL LETTER F	f	0x66		YES
103	0x67	LATIN SMALL LETTER G	g	0x67		YES
104	0x68	LATIN SMALL LETTER H	h	0x68		YES
105	0x69	LATIN SMALL LETTER I	i	0x69		YES
106	0x6A	LATIN SMALL LETTER J	j	0x6A		YES
107	0x6B	LATIN SMALL LETTER K	k	0x6B		YES
108	0x6C	LATIN SMALL LETTER L	l	0x6C		YES
109	0x6D	LATIN SMALL LETTER M	m	0x6D		YES
110	0x6E	LATIN SMALL LETTER N	n	0x6E		YES
111	0x6F	LATIN SMALL LETTER O	o	0x6F		YES
112	0x70	LATIN SMALL LETTER P	p	0x70		YES
113	0x71	LATIN SMALL LETTER Q	q	0x71		YES
114	0x72	LATIN SMALL LETTER R	r	0x72		YES
115	0x73	LATIN SMALL LETTER S	s	0x73		YES
116	0x74	LATIN SMALL LETTER T	t	0x74		YES
117	0x75	LATIN SMALL LETTER U	u	0x75		YES
118	0x76	LATIN SMALL LETTER V	v	0x76		YES

Dec	Hex	Character name	Character	CP1252	Comments	Sup?
119	0x77	LATIN SMALL LETTER W	w	0x77		YES
120	0x78	LATIN SMALL LETTER X	x	0x78		YES
121	0x79	LATIN SMALL LETTER Y	y	0x79		YES
122	0x7A	LATIN SMALL LETTER Z	z	0x7A		YES
123	0x7B	LATIN SMALL LETTER A WITH DIAERESIS	ä	0xE4		YES
124	0x7C	LATIN SMALL LETTER O WITH DIAERESIS	ö	0xF6		YES
125	0x7D	LATIN SMALL LETTER N WITH TILDE	ñ	0xF1		YES
126	0x7E	LATIN SMALL LETTER U WITH DIAERESIS	ü	0xFC		YES
127	0x7F	LATIN SMALL LETTER A WITH GRAVE	à	0xE0		YES

a: This character will look like an new line on the phone.
b: This is as control code, not a character.
c: This character costs 2 bytes instead of 1.
d: This character MUST be URL encoded.

11. SMS Length

The following explains how much data can fit into 1 SMS, when a message will be split and how it will be split.

11.1 The Message Payload

The Message Payload of an SMS is 1120bits. Therefore the contents of an SMS can take up a maximum of 1120bits. We can use these bits in various ways.

11.2 Text SMS (normal SMS)

Each character is 7bits long. So we have: 1120/7=160. There is room for 160 characters in this kind of SMS. MOST characters are 7bits long. A few characters are 14bits long, so using these characters actually counts as 2 characters against the total of 160. Please look at the Accepted Characters section for more information.

11.3 Unicode SMS

This kind of SMS contains text encoded in Unicode. This allows for “foreign” alphabets like Chinese, Japanese, Korean, Hebrew, Arab etc. Each character is 16bits long. So we have: 1120/16=70. There is room for 70 characters in each SMS.

11.4 The UDH

UDH is User Data Header. Some SMS has an UDH, some don't. The UDH is a small amount of data that can be in the beginning of the Message Payload. The UDH informs the handset how to handle the SMS. Since the UDH is inside the Message Payload there is less space left for other things. We must take this into account when we use UDH. The length of the UDH varies but is always a multiple of 8bits.

11.5 Concatenated text SMS

A text message that is longer than 160 characters must be split up into multiple SMS. Each of these SMS must include an UDH that will help the handset put the SMS in the right order. The length of a concat SMS UDH is 48bits (6 bytes).

Each concatenated SMS can hold: 1120-48=1072bits.
Since each character is 7bits long: 1072/7=153.

1 Concatenated SMS can hold 153 characters.
2 Concatenated SMS can hold a total of 2*153=306 characters
3 Concatenated SMS can hold a total of 3*153=459 characters.

If you send a message that is longer than 160 characters Wireless Factory will take care of the splitting, making the UDH etc. and charge you accordingly.

11.6 Concatenated Unicode SMS

The length of a concat SMS UDH is 48bits (6 bytes). Each concatenated SMS can hold: 1120-48=1072bits. Since each character is 16bits long: 1072/16=67 characters. Currently we do not split Unicode SMS that are too long.

11.7 Raw SMS

A raw SMS is an SMS that contains exactly the 1120bits that you want. A raw SMS could include an operator logo, ringtone, contact information (aka vCard), configuration info, calendar event, flash message, WAPpush or many other things. If the handset does not support a particular kind of message it will probably display an error message and delete the message. A raw SMS always include an UDH because without an UDH the handset would not know what to do with the Message Payload. We normally count the length of a raw SMS in bytes so: 1120/8=140. There is 140bytes in 1 raw SMS. If you need more than 140bytes you can put a concat UDH inside a raw SMS to send concatenated raw SMS.

11.8 WAPpush SMS

WAPpush is just a raw SMS with a special UDH. The WAPpush UDH is 56bits. So we have: 1120-56=1064. 1064/8=133bytes. These 133bytes must be WBXML encoded data.

11.9 Concatenated WAPpush SMS

This kind of SMS must include both an WAPpush UDH and a concat UDH. This will take up 96bits. So we have: 1120-96=1024. 1024/8=128bytes. These 128bytes must be WBXML encoded data.

11.10 WBXML (WAP Binary XML)

A WAPpush SMS contains some text and a link. The text is shown on the screen and the user has the option of downloading whatever the link is pointing to. Typically the link points to a soundfile, videofile, imagefile or similar.

Once the file has been downloaded the handset will try to do something with it (play it, store it, etc.). The text and link is described in a special XML format.

Since XML is not designed for low bandwith connections like SMS the XML is incoded into something called WBXML that takes up much less space.

If you send WAPpush via WirelessFactory we will encode the text and link to WBXML and add the necessary UDH. If neccesary we will split the SMS into multiple SMS. You must make sure that the encoded WBXML message is not bigger than what can fit into the number of SMS you are willing to pay for.

A rule of the thumb: if the length of text+link is less than 120 then it will fit into 1 SMS.

12. Rules and regulations

The Danish market is regulated by the framework agreement. All users of shortcodes in Denmark have to observe the regulations in this regulation. The content provider, not the operator or the gateway company has the final responsibility to uphold these rules.

The market is monitored by the Danish Teleankenævn who will fine any trespassers and have authority to shut down any services found to be in violation. Shut down of services only happen in the most severe cases.

In compliment to the framework agreement some of the operators have moral and ethic guidelines. These cover some aspects of doing business that the laws and regulations does not. The gist of them is that if you are in doubt as to the legality of a service then there is no doubt.

12.1 Framework agreement in English:

http://www.rammeaftalen.dk/english/

12.1.1 Price regulations:

http://www.rammeaftalen.dk/english/operational-rules/12-maximum-amounts/

12.1.2 Information regulations:

http://www.rammeaftalen.dk/english/operational-rules/14-information-requirements/

12.2 Rammeaftalen in Danish:

http://www.rammeaftalen.dk/

12.2.1 Price regulations:

http://www.rammeaftalen.dk/rammeaftalen/operativt-afsnit/12-beloebsgraenser/

12.2.2 Information regulations:

http://www.rammeaftalen.dk/rammeaftalen/operativt-afsnit/14-informationskrav/

13. Revision

2.1.1 – KH – 2012.09.24

PHP Example updated and now reflects a §GET post example.